**程序结构设计与分析**

1. **读入地图时使用的是try{}catch方法；初始化地图时，考虑到地形地图和动物地图有覆盖部分，同时打印了地形地图与动物地图，当地形地图与动物地图都为空地时就输出空格，当既有地形又有动物时输出动物，否则输出各自的地图。**
2. 包含了一个子类，目的是对判断可移动的棋子做出移子动作
3. 关键变量：

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量名 | 变量作用 |
| char[][] | tileMap | 存储地形地图 |
| char[][] | leftAnimalMap | 左方动物数组 |
| char[][] | rightAnimalMap | 右方动物数组 |
| char[][][] | copyLeftMap | 左方玩家历史记录 |
| char[][][] | copyRightMap | 右方玩家历史记录 |
| boolean | Player | 用于判断左右玩家顺序 |
| int | currentStep | 用于记录当前步数，以及作为历史数组的下标 |
| int | nextStep | 表示下一步要走到的步数 |
| int | lastStep | 用于记录悔棋前的步数，即最多能取消悔棋回到的步数 |
| int | undoCount | 用于记录连续悔棋的次数 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

4.函数：

|  |  |  |
| --- | --- | --- |
| 函数 | | 作用 |
| static void enter(boolean player, Scanner scanner) | | 进入手动输入操作区 |
| static void assistFunction(String input, boolean player) | | 一些基础操作的判断与执行：悔棋、取消悔棋等 |
| static void eatingFunction(String input, boolean player) | 吃子系列判断与执行 | |
| static String[] checkLeftMoveToLeft(boolean player, int i, int j)  系列 | 判断吃子系列操作是否可以进行并返回须执行的步骤（所有步骤已经编序） | |
| static void WInOrLose(boolean player) | 判断是否有玩家胜利 | |
| static void save() | 存储历史记录 | |
| static void printMap() | 打印地图 | |

**5.觉得写得好的地方是悔棋与取消悔棋，不好的地方就是代码太长，好多地方代码有重复却改不了。**

**编程中遇到的问题和解决策略**

### 遇到的问题（解决方法）

1. 一开始的时候卡在打印的地方，不知道怎么输出一个完整的可以辨别左右方的地图 （最开始的时候，我把助教给的地图改成abcd…与1234…的结合来判断是左方动物还是右方动物，并且成功了但遇到了下一个问题）
2. 不知道该怎么判断左右方吃与被吃的关系 （于是我又把原本的一个动物地图拆分成左右方独立的两个地图，都用123…来表示动物，然后就容易判断了（后来想到其实abcd…可以用减去一个内码来与123…比较判断大小的，但那时都快完工了））
3. 把所有的操作写在了一个方法里，基础功能还好，到最后就难以判断五子可动了（把那个方法拆分成多个子方法，并且在能否落子的方法处给了一个返回值，在最后的时候就可以通过这个返回值是否改变来判断无子可动了）
4. 不会debug （看助教的教学视频，但感觉还是不怎么会用）

### 意见与建议

可以多出一些关于如何使用与Java相关的App的教学视频